EXHIBIT E

FILED UNDER SEAL

CLEMENT SETH ROBERTS (STATE BAR NO. 209203)
croberts@orrick.com
BAS DE BLANK (STATE BAR NO. 191487)
basdeblank@orrick.com
ALYSSA CARIDIS (STATE BAR NO. 260103)
acaridis@orrick.com
EVAN D. BREWER (STATE BAR NO. 304411)
ebrewer@orrick.com
ORRICK, HERRINGTON & SUTCLIFFE LLP
The Orrick Building
405 Howard Street
San Francisco, CA 94105-2669
Telephone: +1 415 773 5700
Facsimile: +1 415 773 5759

SEAN M. SULLIVAN (*pro hac* vice)
sullivan@ls3ip.com
MICHAEL P. BOYEA (*pro hac* vice)
boyea@ls3ip.com
COLE B. RICHTER (*pro hac* vice)
richter@ls3ip.com
LEE SULLIVAN SHEA & SMITH LLP
656 W Randolph St., Floor 5W
Chicago, IL 60661
Telephone: +1 312 754 0002
Facsimile: +1 312 754 0003

Attorneys for Sonos, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

GOOGLE LLC, Plaintiff and Counter-defendant, v. SONOS, INC., Defendant and Counter-claimant.	Case No. 3:20-cv-06754-WHA Related to Case No. 3:21-cv-07559-WHA **OPENING EXPERT REPORT OF DOUGLAS C. SCHMIDT**

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1   Dep. Tr., 122:2-124:4.

2   **4.   Receiver Playing Back Retrieved Media**

3   162.   After receiving the setPlaylist message, the Receiver functions to retrieve and

4   playback a media item (e.g., the media item corresponding to the videoId in the setPlaylist

5   message). *See, e.g.*, Mo Dep. Tr., 142:6-22; GOOG-SONOSWDTX-00037243 [MDX

6   Communication Protocol v3], 51-52 ("Server to Screen" setPlaylist message containing "videoId"

7   for "*[v]ideo that should start playback ASAP*."), 58-59 ("Remote to Server" "setPlaylist" message

8   containing "videoId" for "*[v]ideo that should start playback ASAP*."); GOOG-SONOSWDTX-

9   00041491 [YouTube Orbit (MDx) Android - MDx Playback on Android], 95:

10   Eventually ███████████████████ (MdxPlaybackDescriptor initialPlayback)
         gets called with the MdxPlaybackDescriptor. This will set the information of
11   MdxPlaybackDescriptor into 'setPlaylist' params outlined here and send this with
         the connection properties when connecting. In response, the server will initialize the
12   RQ playlist with the given playback information and ***tell the TV to start playing the***
         video.
13
         163.   In this regard, the Receiver uses a given videoId (e.g., from the setPlaylist message)
14
     to retrieve a corresponding media item from the given YouTube service. In particular, the Receiver
15
     makes a "getPlayer" call with the videoId to the "Player Service" in the YouTube cloud
16
     infrastructure and receives a response containing one or more URLs for the media item. *See, e.g.*,
17
     Mo. Dep. Tr., 137:17-25, 138:13-23 ("At a higher level, the receiver, when it has a video ID, what
18
     happens … for playback typically contains a get player requests [sic]. That's a network request.
19
     And then that response contains information that is needed."), 139:13-140:1 (**Q:** "Is the get player
20
     request targeted to a service that is external of the receiver device?" **A:** "It hits the player service,
21
     which is a back end."), 141:8-15, 142:6-144:15 ("From my perspective, … I guess the receiver or
22
     client gets a video ID and … what happens after that is a get player request is made and then the
23
     response … it may contain, you know, URLs related to where the content may be."), 146:1-7;
24
     Nicholson Dep. Tr., 93:23-94:11, 113:15-115:17, 124:24-125:11; *see also, e.g.*, GOOG-
25
     SONOSWDTX-00039491 [YouTube Orbit (MDx) - Credential Transfer Tokens (CTTs)], 91
26
     (illustrating Receiver making "GetPlayer" request with videoId to "Player Service"); GOOG-
27
     SONOSWDTX-00039785 [Server], 85-86 (describing and illustrating process by which "RPC to
28
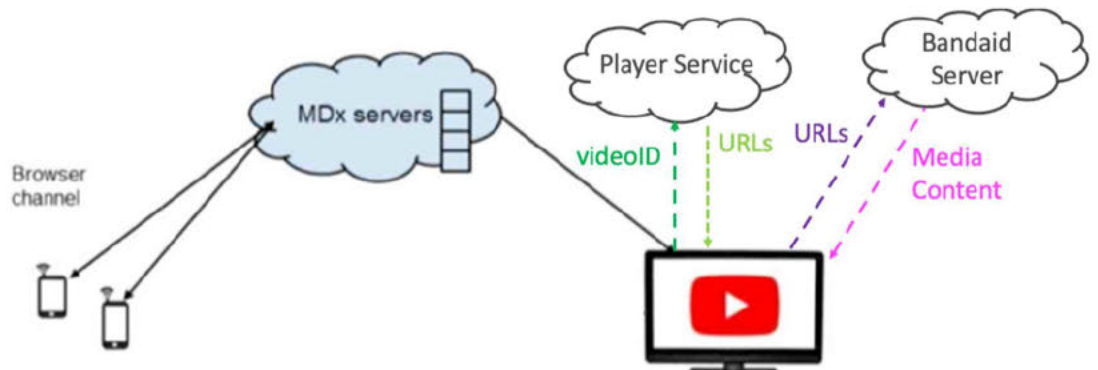   GetPlayer for available streams URLs" is made), 89; GOOG-SONOSWDTX-00052121 [YouTube

Developer's Handbook – Life of a video watch], 23-26; GOOG-SONOSNDCA-00115814 [Edge Streaming and Bandaid CDN], 15, 20.

164.   The Receiver then uses the one or more URLs to retrieve the media item from one or more "Bandaid" servers in the YouTube cloud infrastructure for the given YouTube service and then render the retrieved media item at the point within the media where the Sender was rendering when it detected the selection of the Cast button/icon and the selection of the Receiver from the Device Picker menu. *See, e.g.,* Mo Dep. Tr., 142:6-144:15 ("YouTube playback relies on what we call band-aid servers…. [T]hey are kind of like a cache system for storing … video bits for playback. As far as I understand, player responds and obtains information that the player needs to know … where these servers or caches are to load from."), 144:19-145:6, 145:12-25; Nicholson Dep. Tr., 65:16-67:11, 113:15-115:17, 124:24-125:11; *see also, e.g.,* GOOG-SONOSWDTX-00052121 [YouTube Developer's Handbook – Life of a video watch], 23-26; GOOG-SONOSWDTX-00039785 [Server], 85-86 (describing and illustrating process by which "RPC to bandaid for audio or video content" is made).

165.   Notably, the aforementioned process by which a Receiver plays back media content occurs whenever the Receiver is to playback a media item, such as after finishing playback of the initial media item for the Cast session, after a user skips forward to a next media item, etc. *See, e.g.,* Mo Dep. Tr., 169:6-19, 175:1-23. For instance, after playing back the initial media item, the Receiver performs the aforementioned process using a videoId for the next media item contained in a received WatchNextResponse from the WatchNext service in the YouTube cloud infrastructure.

166.   The process by which a Receiver plays back a media item corresponding to a videoId

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

is shown below in a demonstrative that builds on Google's MDx architecture diagram:

GOOG-SONOSWDTX-00037264 [Orbit & Cassini Deep Dive], 69 (modifications added); *see also, e.g.*, GOOG-SONOSWDTX-00050334 [Playback params passing in Audio Tier], 40-41 ("10. Client app [on Receiver] calls InnerTube getPlayer with the video ID and playlist ID…. 19. GetPlayer returns the streamingUrl … to the client…. 20. YTM app [on Receiver] calls GetWatchNext with video ID and playlist ID…. 25. GetWatchNext returns the next track in the response.").

5.   YouTube Sender in Casting Mode of Operation

167.   After detecting the selection of the Cast button/icon and the selection of a particular Receiver (or set of Receivers) from the Device Picker menu, the Sender functions to detect an indication that playback responsibility has successfully transferred from itself to the selected Receiver (or Receivers), after which the Sender switches to operate in the Casting mode. *See, e.g.*, GOOG-SONOSWDTX-00041491 [YouTube Orbit (MDx) Android - MDx Playback on Android], 95 (describing YouTube Sender transitioning its "Player stack" from a "local" mode to an "MDx or 'remote' mode" after detecting "MdxSessionStatusEvent").

168.   The Sender detecting this indication and transitioning from the non-Casting mode to the Casting mode in which the Receiver (instead of the Sender) is configured for playback of the Watch Next queue is evident from changes to the user-facing appearance and functionality of the Sender. For instance, the appearance of the Cast icon displayed by the Sender is changed to indicate that the Sender is now connected to a Receiver, thereby informing the user that a Receiver is now responsible for playback as opposed to the Sender. *See, e.g.*, SONOS-SVG2-00060294 [Play media from Chromecast-enabled apps to your speaker or display], 94 ("When you're connected, the Cast button will turn from light to dark grey, letting you know that you're connected."); GOOG-SONOSNDCA-00061126 [Cast audio from Chromecast-enabled apps to speakers] ("The Cast button will change colors, letting you know you're connected."); SONOS-SVG2-00060340 [Watch YouTube Kids videos on your TV] ("Once connected, you can start watching [on your Chromecast, game console, or smart TV] when the Cast icon turns blue"); GOOG-SONOSWDTX-00038270,